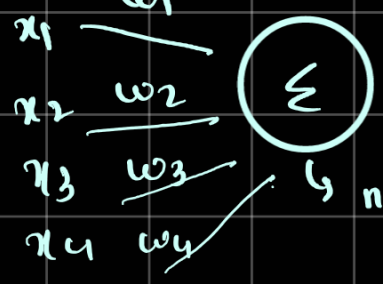
Logistic regression → linear combination → sigmoid




↳ neuron

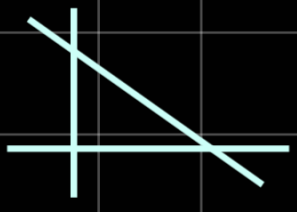
when we train a model

↓

$y = mx + c$

if $c \neq 0$

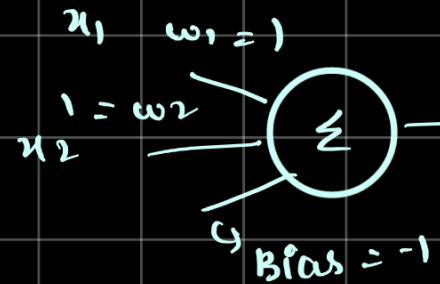
Boolean AND



| $x_1$ | $x_2$ | $y$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |


↳ Bias values

→ we take values of $w_1$, $w_2$ and Bias values to replicate the results of Boolean gates




Each bubble here is called a neuron and the functionality of logic gates can be replicated using singular neurons

↳ Here we are able to seperate data through a line, thus the data is called linearly seperable.

↳ Thus, the data that is linearly sperable can be solved using a single neutron

↳ But when we have to solve XNOR we need a neural network cause it is a combination of two logic gates; two neurons;

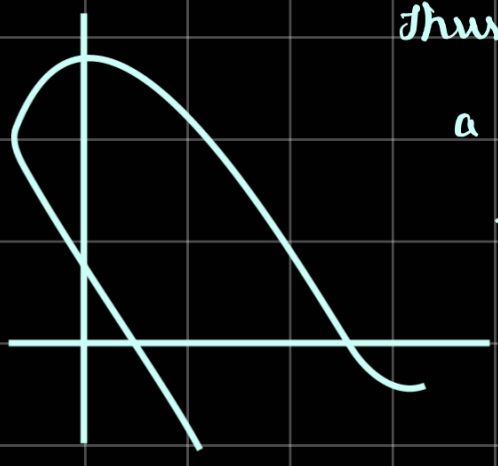
a neural network

↓

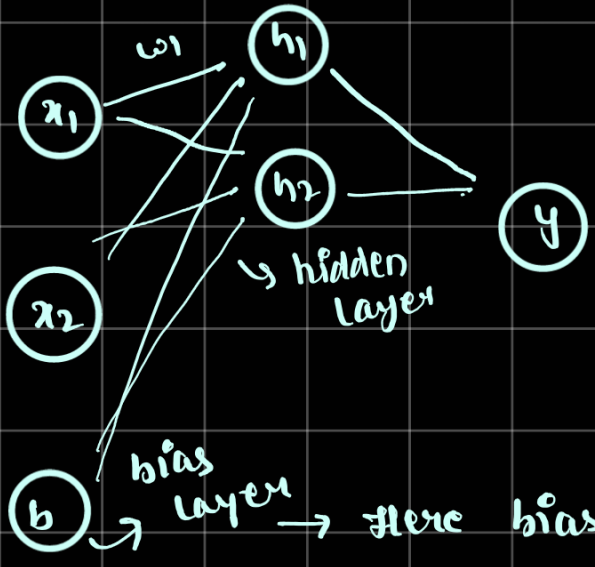
a non-linearly seperable curve

↓

| $x_1$ | $x_2$ | y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |



Thus, we use a hidden layer apart from simplistic neurons to solve.




→ Here bias is considered -1.